conventional SMBIOS programs. Specifically, the apparatus and methods of the present invention provide a template file intermediate between the utility program and database of SMBIOS. The template file includes all information required for interpreting and displaying the data structures stored in the SMBIOS database. In this regard, when the utility is commanded to retrieve and display data from the SMBIOS database, it retrieves the data from the SMBIOS database. Based on the Type and offset associated with the data, the utility program accesses the corresponding information for interpreting and displaying the data from the template file. Using this information, the utility program then properly interprets and the displays the requested information concerning the computing system.

Importantly, because the information for interpreting and displaying the data stored in the SMBIOS database is removed from the source code of the utility program and placed in the template file, updates to this information is performed in the template file, as opposed to the utility program's source code. Thus, as the SMBIOS database is updated to new versions, the utility program does not necessarily require updating. Instead, updates for interpreting and displaying can be performed in the template file and new versions of the template file created. In this regard, the utility program and all versions of the template file may be stored in the computing system. The utility program will determine the version of the SMBIOS database resident in the computing system and select the corresponding template file version to interpret and display data stored in the SMBIOS database. As such, all of the computing systems within a network will use the same version of the utility program, but their utility programs will individually select the proper version of the template file to use with the corresponding version of the SMBIOS database.

As an example, in one embodiment, the present invention provides a system management apparatus for retrieving and displaying SMBIOS data relating to the configuration and components of a computing system to a user via a display terminal. The apparatus includes a database of SMBIOS structures stored on a computer-readable medium. Each structure is divided into offsets called Fields, where each Field contains data related to a particular aspect of the configuration of the computing system or components of the computing system. Further, the apparatus includes a utility program

stored on a computer-readable medium, which in response to commands from a user, retrieves data from the SMBIOS database and displays the data on the display terminal. Importantly, the apparatus also includes a template file stored on a computer-readable medium separate from the utility. The template file contains information for interpreting and displaying the data retrieved by the utility program from the database. As such, the template file eliminates the requirement that the utility program include all of the strings, field descriptions, bit definitions, etc. for interpreting and displaying the data stored in the SMBIOS database.

With regard to the configuration of the template file, data is stored in the SMBIOS database in various manners. For example, data may be stored as raw data values, strings, free form strings, enumerated values, bit fields, or grouped bits. The information for determining the data's structure, its content, and any descriptive text associated with the data is stored in the template file. In operation, the utility program, using parsing routines and standard SMBIOS calls retrieves data from the SMBIOS database. The utility program then uses the template file to determine the structure of the data and what information the data provides, based on the Type and offset associated with the data as stored in the SMBIOS database. The utility then displays the interpreted data along with related text information to the user.

To interpret the data structures stored in the database, the template file of the present invention includes several different types of descriptor keys. These descriptor keys describe the purpose of a particular offset, or Field, in an SMBIOS data structure stored in the SMBIOS database. For example, one descriptor key referred to as DATA_ID is used to inform the utility that the data stored in the Field and associated with the descriptor key is raw data. Examples of raw data values are clock speeds, memory speeds, handles, etc associated with the computing system. The DATA_ID descriptor key includes information relating to the length of the data field and the format for displaying the raw data, such as hexadecimal, decimal, etc. The DATA_ID descriptor key also includes text to accompany the data when displayed. For each raw data Field in a particular structure Type in the SMBIOS database, the template file includes a corresponding Field definition that uses the DATA_ID descriptor key to define the structure of the offset and provide text to be displayed with the data.

In some instances, the data stored in the SMBIOS database are text strings. For this type of data, the template file of the present invention includes a descriptor key, referred to as STRING_ID, indicating to the utility program that the data for the current Field in the Type data structure retrieved by the utility program is a string. The STRING_ID descriptor key, in addition to identifying the data as a string, also includes text to be displayed with the string, thereby providing additional information concerning the contents of the string. For example, if a string Field in an SMBIOS structure defines the name of the vender for the BIOS software, the template file of the present invention will include a STRING_ID descriptor key associated with this Type and offset in the SMBIOS database. The STRING_ID descriptor key will indicate that the data is a string. It will also include text associated with the descriptor key such as "BIOS Vender's Name" to be displayed with the data string for the BIOS vender's name.

Bit field structures may also be used in the database, where the individual bits of the bit field represent yes/no or true/false states about, features, components, or settings within the computing system. For example, associated with each bit in a bit field may be a setting for a particular component of the computing system. If the bit is set to true, then the setting is true of the component, but if set to false, it is not true. For these bit field structures, the template file of the present invention includes a BIT_FIELD_ID descriptor key. The BIT_FIELD_ID descriptor key first identifies the Field stored in the SMBIOS database as a bit field and will provide parameters for interpreting the bit field. It also provides text to be printed describing the contents of the bit field, as well as text to be displayed depending on whether a bit in the bit field is a 1 or a 0. It also includes text to be displayed for each bit describing what the bit represents.

Enumerated values may also be present in the SMBIOS database, where different values represent different features, components, or settings within the computing system. Specifically, one possible value of the enumerated value is one setting, while another possible value represents another setting. For this type of data structure, the template file includes a corresponding descriptor key referred to as ENUM_ID. The ENUM_ID key includes different text to be displayed based on the value of the enumerated value, such that depending on the value of the enumerated value appropriate text will be displayed to the user. In this embodiment, the utility program initially retrieves the enumerated value